TITLE:    METHOD FOR EFFICIENTLY DOWNLOADING SCSI AND
          SERVO FIRMWARE TO SCSI TARGET CONTROLLERS

## FIELD OF THE INVENTION

This disclosure involves enhancements in the process of loading microcode from a system to a target controller for a disk or tape drive.

## BACKGROUND OF THE INVENTION:

Many of the modern computer system networks utilize and require that specialized firmware be downloaded for placement on a target medium such as a
5 disk or tape drive.

Figs. 1A and 1B are illustrative of arrangements for downloading firmware in a SCSI (Small Computer Systems Interface) environment. Many inherent problems occurred in such downloading operations, such as
10 properly identifying the target device and properly selecting the appropriate firmware to be downloaded. Then also, a latency problem occurred when the buffer arrays were not of sufficient size to handle the large volume of firmware bytes required for the download thus
15 requiring sequences of multiple download commands to accomplish the download. It should here be understood that the target for download of firmware is the Controller (Peripheral Controller) which holds the microcode which is used to handle and control a
20 peripheral device such as a disk, a set of disks, or tape unit.

A basic function of computer systems and digital modules very often is involved with loading microcode into various digital devices to make them
25 properly operative. In many earlier systems SCSI firmware could only be downloaded using a single download command together with a single dimensional buffer array. On many occasions, the downloading utility program was faced with a situation where the system array capacity
30 was not sufficiently large enough to accommodate the firmware that was required to be downloaded.

As a result, the downloading was delayed and had to be provided in a series of subsequent steps before the proper amount of firmware could be provided to the
35 target digital modules.

awk\appl\041468L.doc

The present system uses a library-exported interface to issue command requests. Then by adding a SCSI Inquiry command and querying the target device with information from a Page Code field, it was possible to

5 retrieve the firmware numbers page. Thus the information returned contained the appropriate firmware release numbers and the servo RAM release numbers which the utility program could integrate this information in order to download the correct firmware.

10 The present disclosure provides a special SCSI Inquiry command which provides required information to produce access to the correct firmware, which can be downloaded in multiple cycles through the offices of a selective single or dual system array capacity, which

15 overcame the limitations and delays caused by the earlier systems.

Figure 1B shows the basic relationship between different software drivers and how they are utilized in the overall operation of a SCSI environment. Normally

20 the Device Drivers 74, 76, 68, etc. will reside in the central processing unit's RAM 73 and they are loaded via the systems configuration utility program during the boot operation. In the disk operating system (DOS) environment, a program CONFIG.SYS provides a host

25 computer with its operational parameters. The HA (Host Adapter) driver 82 resides in the host adapter 80 usually in some type of memory, such as EPROM.

The operating system 72, which calls a target driver, connects the various application programs 70a,

30 70b, 70c, over to either the tape driver 74, the disk driver 76 or the CD ROM driver 78. These then connect to the host adapter (HA) 80 and the HA number 1 driver 82. The HA driver requests command control blocks (CCB's) from the RAM 73 in the CPU. The data is then conveyed on

35 SCSI bus 37 over to the peripheral controller 40, which has the controller firmware 86 which has to be loaded.

awk\appl\041468L.doc

The small computer systems interface (SCSI) uses the concept of host memory blocks for Command, Data, and Status, which will be interchanged between the host computer and the SCSI device. The host memory blocks are
5    prepared by the host and managed by the host adapter. These host memory blocks must actually be found by the peripheral device which is attached to the host adapter. This is accomplished through what is known as "Pointers".

Figure 1C shows the basic physical components
10   of a SCSI based system. These are items which include the "Initiators" which involve usually host adapters, (HA), that is to say, the devices that initiate an I/O process. Then there are the modules which involve the peripheral controllers, that is to say the devices that
15   control the I/O process and are shown in the lower portion of Figure 1C. There are also the Logical Units which are the physical peripheral devices, that is to say, modules where the data is stored or retrieved, and in addition there is the SCSI bus, which is the
20   connection between the initiators and the targets.

Thus, Figure 1C shows how information is transferred from the application program 70 over to the physical target device 96. Figure 1C shows the application program 70 which is applied to the operating
25   system 72, which connects to the target device driver 75x. Here there is a two-way communication between 75x and the host adapter driver 82, which then connects to the SCSI chip 86. This first set of modules involve the host system and from thence the connections go on the
30   SCSI bus 37, showing the command status, the message and monitor bus phases, and the arbitration selection re-selection transfer data. Thus the SCSI chip 86 connects to the SCSI chip 88 which loads the firmware 90. The SCSI chip 88 also loads the buffer 92, which is connected
35   to the drive electronics 94, which operates the physical target device 96.

awk\appl\041468L.doc

The present invention provides a faster and more efficient method for downloading firmware to a selected peripheral controller while selecting appropriate firmware for a target medium and enabling the download to occur with a minimum of sequential commands and often accomplishing the download in just one download command.

## SUMMARY OF THE INVENTION:

The present system involves a method for downloading large volumes of SCSI firmware and servo firmware from an operating system over to SCSI devices using a set of two specialized two-dimensional arrays to overcome the limitations of systems array capacity.

Earlier utility programs used for downloading SCSI firmware involved a single dimensional buffer array which utilized a single download command. The presently developed method utilizes a library-exported interface to issue multiple command requests. In addition, it adds a SCSI Inquiry command to query the target device with information from a Page Code field where it was then possible to retrieve the firmware page number.

This information then provides appropriate firmware release numbers and servo RAM release numbers which the ongoing utility program could integrate in order to download the correct firmware and provides for the download in a single cycle through the use of dual two-dimensional system capacity arrays. Thus, extremely large volumes of firmware can often be downloaded in a single command cycle and the system provides reliability by enabling a checkout to occur that the proper firmware has been downloaded.

A selection is made as to use of a single dimensional array or a dual two-dimensional array for buffer loading. Further, check-sequences are provided to determine whether the download of firmware is merely the same as the already-installed firmware or whether the new firmware is new and different from the priorly loaded firmware.

awk\appl\041468L.doc

## BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1A is an overall system drawing showing the environment in which the present method can be implemented; Figure 1D is a drawing showing the use of

5 different software drivers in a SCSI environment; Figure 1C is a drawing showing how information is transferred from an Application Program to the physical target device; Figure 1D is an overall operational diagram for a disk control target;

10 Figure 2 is a sketch showing a typical example of a Command Descriptor Block;

Figure 3 is a drawing of the Write Buffer Command used in conjunction with the Read Buffer Command, which can be utilized as a diagnostic function for

15 testing the integrity of the delivery system;

Figure 4 is a diagram showing the Write Buffer Mode field;

Figure 5 is a drawing which displays the elements of the Operation Code;

20 Figure 6 is a drawing illustrating a two-dimensional array which can provide the buffering arrangement for firmware which is to be downloaded to a SCSI target control device;

Figures 7A, 7B, 7C and 7D are flow charts

25 illustrating the steps involved to implement the download operation to SCSI target control devices.

## GLOSSARY OF RELATED ITEMS

BUFFER ARRAY: - is a temporary area of memory used to store data when issuing a SCSI Read or Write command. This array can be a multiple dimensional array.

5    BUFFER ID: - This is a field of data which identifies a specific buffer within the Logical Units. The buffer ID is assigned by the Vendor to identify different buffers if more than one is used during the data transfer. The reason for using more than one buffer is for efficiency.

10   For example, if a user is using two buffers while the first buffer is being used to transfer the first data block, the second one is being loaded with the next contiguous offset data block.

BUFFER OFFSET: - is a scheme to facilitate skipping over

15   data that has already been transferred with a Read/Write command and provides an offset from the start of the buffer to indicate where the new data transfer will start.

CDB: - stands for Command Descriptor Block. A SCSI

20   command is executed by sending a Command Descriptor Block to the target. The first byte of the CDB is the Operation Code which has a Group field which describes the number of command bytes in the CDB. The lengths can be six, ten or twelve bytes. It also has the Command

25   Code for example 3B (hex) for the Write buffer command. The other major fields in the CDB are LUN (Logical Unit Number) which is the LUN of the target device; LBA (Logical Block Address) field which is the address on the device where the transfer will start/commence such as

30   during a Read or Write operation. Another important field is the Transfer Length Field which specifies the amount of data to be transferred, usually the number of blocks. These blocks can all be the same length or of variable length. A CDB may also have a Parameter List

35   Length Field which specifies the number of bytes sent out

awk\appl\041468L.doc

with a particular parameter. The last byte in a CDB is the Control Byte Field which is vendor specific.

COMMAND SEQUENCE ERROR: - This is when an improper sequence of software commands occur It is detected by

5 the Sense Code.

CONTROLLER: - is the hardware and firmware that control the disk or tape device. There may be many devices or a single device under the control of the controller. This firmware normally resides in the flash prom. It can be

10 updated or replaced by a download software utility program.

CONTROLLER INQUIRY ATTRIBUTES: - is the data received after an inquiry of the Controller. The following is an example of the data returned by this command: Vendor

15 identification, data identifying the vendor of the product such as Unisys; Product Identification, data defined by the vendor to identify the product; Product Revision level, data as defined by the vendor to identify the revision of the product. This information is used by

20 the download utility program to verify and insure that the proper firmware is being downloaded to the device.

CONTROL MEMORY SPACE (of the Logical Unit): - is the ROM memory in the Logical Unit where the Control Program/firmware resides.

25 DATA-OUT-BUFFER: - is the memory buffer area used to temporarily store the data for an outbound data transfer such as in a WRITE command.

DFAST UTILITY PROGRAM: - (Download For All SCSI Targets) is a utility program that downloads microcode/firmware to

30 SCSI devices such as controllers for disk drives and tape drives attached to a Unisys A-Series computer system.

EVPD: - (Enable Vital Product Data) is a bit in the SCSI Inquiry command that when set to a logical "1" it specifies that the target/device shall return the vital

35 product data specified by the page code field. When this

awk\appl\041468L.doc

bit is set to "0", the target shall return the standard Inquiry data.

EVPD SET: - is bit 0 located in byte 1 of the Inquiry command called the Enable Vital Product Data bit. When

5 this bit is set to a logical "1" it specifies that the target/device shall return the vital product data specified by the page code field. When this bit is set to "0" zero the target shall return the standard Inquiry data.

10 FIRMWARE HEADER: - is an 8192 byte header appended to the front end of the firmware that standardizes the format of the firmware file which will simplify the task of reading the firmware file by the system application program. Some of the specified data required to build the header

15 are: Customer ID, Size of header record, Size of firmware data, New firmware level, Vendor ID, Number of drive ID, Number of replaceable firmware levels, Firmware levels, and Drive IDs.

FIRMWARE NUMBERS PAGE: - is the name of the page that

20 provides the firmware version numbers.

FIRMWARE RELEASE NUMBERS: - is a scheme used to identify a particular version or level of firmware that has been released to the customer. It can be an initial firmware release or an updated firmware release which provides a

25 fix or feature to the firmware. It is also the firmware release number returned in page C0h (bytes 4 thru 11) when the EVPD bit is set in the Inquiry Command.

GOOD STATUS: - Means that the command that was issued was successfully completed or that the requested operation is

30 satisfied. This occurs when the initiator (CPU) receives "ooh" in the Status Byte Codes. A Status Byte is sent from the target/peripheral controller to the Initiator during the STATUS phase at the completion of each command (unless the command is terminated by an ABORT message,

35 reset condition or an unexpected disconnect).

**ILLEGAL REQUEST:** - Indicates that there was an illegal
parameter in the command descriptor block or in the
additional parameters supplied as data for some commands
or that the command being issued is not supported by the
5   device.

**INQUIRY COMMAND:** - This is an ANSI SCSI command which was
sometimes called the Attributes Command.  This command
has the EVPD bit set to Enable Vital Product Data that
contains the Servo Revision information.  This is a
10   command from the CPU via a Channel Adapter which was sent
to the target device (controller) to get information
about the vendor, the Product ID, and Product Revision
Level.  See also CONTROLLER INQUIRY ATTRIBUTES.

**LOGICAL UNITS:** - Are usually assigned to unit-groups such
15   as peripheral devices.  Each group has one or more
logical units, beginning with logical unit "0" up to a
maximum of eight Logical Units.  A Logical Unit is the
device, such as a disk or tape, attached and controlled
by a target/peripheral controller.  These Logical Units
20   can have a 1-to-1 relationship with a physical drive or,
if in a RAID configuration, may have 2 or more physical
disks per Logical Unit.

**LOGICAL UNIT BUFFER:** - Is the temporary or buffer storage
area in RAM used by a logical unit to temporarily store
25   data before it is written to permanent storage such as
disk, tape, or PROM during a read-write command.

**MODE CONTROL:** - As seen in Figs. 3 and 4, are SCSI
commands that allow downloading of data depending on the
value of the Mode field in the Write Buffer command.
30   Bits 0 through 2 of byte 1 of the Write Buffer command
defines the Mode Control.  A value of 4 is download
microcode to the target.  A value of 5 is download
microcode and save.  A value of 6 is download microcode
with offsets.  And a value of 7 is download microcode
35   with offsets and save.

**MASTER CONTROL PROGRAM (MCP):** - The operating system developed for the Unisys Corporation A-Series computer system platforms.

**ONE DIMENSIONAL ARRAY:** - For example A[0:9] is a one

5 dimensional array with 10 elements. It can hold 10 items or numbers. The first number inside the brackets (0) is the lower boundary and the second number inside the brackets (9) is the upper boundary.

**PAGE CODE FIELD:** - specifies which page of vital product

10 data information the target/device shall return. The information in this page is usually firmware version numbers.

**READ INQUIRY:** - Read Inquiry is used to read the Vendor Identification, Product Identification, Product Revision

15 level and Vendor-Specific data. This information is used to verify that the firmware download is the proper firmware for this device. This is accomplished by comparing the Read Inquiry data with the information provided in the firmware header file.

20 **READ BUFFER DESCRIPTOR:** - The READ BUFFER command is used in connection with the WRITE BUFFER command as a diagnostic for testing target memory and SCSI bus integrity. The READ BUFFER DESCRIPTOR is defined by the Mode Field of the READ BUFFER command byte 1, bits 0, 1 &

25 2. When bits 0 & 2 are set, the READ BUFFER DESCRIPTOR returns 4 bytes of data. Byte 0 is the Offset Boundary alignment within the selected buffer for subsequent WRITE BUFFER and READ BUFFER commands. Bytes 1, 2 & 3 is Buffer Capacity, and returns the size of the selected

30 buffer in bytes.

**RECOVERY PROCEDURE:** - Is a procedure used by a target/device to recover from a data error during a read/write operation. This procedure may entail offsetting the head position from the track center,

35 reallocating of the defective data blocks, or through a

awk\appl\041468L.doc

software algorithm such as the ones used in RAID configurations.

SCSI INQUIRY COMMAND: - is used by a system to determine the configuration of the SCSI device. Target devices
5   respond with information that includes their type and standard level and may include the vendor's identification, product identification model number, product revision level and other useful information, such as vendor specific data.

10   SCSI FIRMWARE: - is firmware that interprets and carries out the SCSI commands. It makes sure that the data is properly written or read and provides recovery procedures when errors are encountered.

SENSE KEY: - is a 4 bit code, bits 0-3 of byte 2 of the
15   sense data received after a Request Sense command that describes an error or exception condition.

SERVO FIRMWARE: - is firmware that positions and controls the tracking of the head assembly over the data tracks on a disk drive to insure proper data during read or write
20   operations.

SERVO RAM RELEASE NUMBER: - is the release level of the Servo RAM firmware. This data is returned in page C0h bytes 12 thru 19 when the EVPD bit is set in the Inquiry Command.

25   TARGET INQUIRY DATA: - is the information provided by the Inquiry command from either a disk or tape device. Some of the information returned is Vendor Identification, Product Identification, and Product Revision level.

TEST UNIT: - This is a command that allows an Initiator
30   to poll a Logical Unit (LUN) to recognize when a Logical Unit is ready to be accessed without any need to allocate space for returned data. Sometimes the firmware can transfer data transmission from a SCSI narrow mode to SCSI wide mode. The Test Unit command can use a re-
35   selection of the transfer mode by updating the properties of the Logical Unit.

awk\appl\041468L.doc

**TWO-DIMENSIONAL ARRAY:** - For example A[0:9,0:19] is a two-dimensional array. The first set of numbers (0:9) inside the brackets is the number of rows which is 10 and the second set of numbers (0:19) inside the brackets is

5 the number of elements in each row which is 20.

**UNIT ATTENTION CONDITION:** - indicates that the removable medium may have been changed or the target has been reset. A target/peripheral controller will generate a Unit Attention condition for each Initiator on each

10 Logical Unit whenever the Target has been reset by a hard reset, power-on reset, or Bus Device reset. In this way, the Logical Units can inform the target/peripheral controller that they are up, ready and operational.

**USERMAINTREQUEST:** - is an MCP (Master Control Program)

15 software module that provides an interface to the DFAST utility program in order to perform its task of downloading firmware.

## DESCRIPTION OF PREFERRED EMBODIMENT

Referring to Figure 1, there is seen the overall operating environment in which the present system for enhanced downloading of firmware (microcode) is

5  accomplished in a rapid and efficient manner.

The web firmware 10 is seen being downloaded to a workstation 20 which adds a header to the firmware involved. This is then conveyed to the central processing unit 30 which uploads the firmware to the

10  central processing unit's main memory 35.

*Subai*  The firmware is then passed to the peripheral controller 40d which can then load this on to the disk drive 50.

Likewise, the firmware in the memory 35 can

15  also be loaded to a peripheral controller 40t to the tape drive 60.

The disk drive 50 and the tape drive 60 are representative of a series of logical units (50n, 60n) which may be viewed as a set of multiple modular units.

20  Referring to Fig. 1D, there is seen an overall system diagram to illustrate the functional operation of the present invention. For example, there is seen in Fig. 1D, a World Wide Web 10 which can supply firmware which can be downloaded into a CPU memory 35 which is

25  operated by the CPU 30. The memory 35 will be seen to have a microcode firmware for SCSI 35f, and the servo firmware SCSI 35s. This information can be temporarily loaded into disk drives 50a and 50b for temporary storage purposes.

30  Subsequently, the peripheral controller 40 will communicate with a flash PROM 76d and a flash PROM 76s. These function to place the SCSI firmware in the flash PROM 76d, and the servo SCSI firmware in the servo flash PROM 76s. The flash servo PROM firmware will provide the

35  head positioning for the Read/Write heads over the target

awk\appl\041468L.doc

disk 50t while the flash PROM 76d will load the data through the Read/Write heads on the opposite side of disk 50t.

5   Referring to Fig. 2 there is shown a typical example of a Command Descriptor Block. In this case, the illustration shows a 12-byte block. Thus a SCSI command is executed by sending a Command Descriptor Block to the target, such as the logical units 50 or 60 of Fig. 1. The first byte of the Command Descriptor Block (CDB) is

10  the Operation Code which has a Group field which describes the number of Command bytes in the CDB. For example, the byte sizes can be 6 bytes, 10 bytes, or 12 bytes, which is the example shown in Fig. 2.

The Operation Code of Fig. 2 is made of two

15  fields shown as the Group Code and the Command Code (as illustrated in Fig. 5). The other important fields of the CDB of Fig. 2 are the Logical Unit Number (LUN) which is the logical unit number of the target device. Further, the CDB has a Logical Block Address (LBA) field,

20  which is the address on the target device where the transfer will start or commence such as during a Read or a Write operation. Another field of the CDB at Fig. 2 is the Transfer Length Field which specifies the amount of data to be transferred, usually as a number of blocks.

25  The blocks can be all of the same length or of a variable length. The CDB may also have a Parameter List Length Field which specifies the number of bytes sent out with a particular parameter. The last byte in the CDB is the Control Byte Field which is specific to a given vendor.

30  Thus, as a result of the head positioning firmware and the Read/Write heads which can receive the firmware from the World Wide Web 10, these can now be placed on the final physical target disks, such as disk 50t.

35  Figure 3 is a drawing of the Write Buffer command which is used in conjunction with the Read Buffer

command as a diagnostic function for testing logical unit memory in the target SCSI device and for testing the integrity of the service delivery subsystem. There are additional modes which are provided for downloading
5   microcode and also for downloading and saving the microcode.

In Figure 3, the horizontal top portion shows the sections indicating seven bits while the vertical column at the left indicates a total of 10 bytes (0-9).
10   The byte "0" indicates the Operation code (3Bh). The byte "1" is reserved at bits 3 thru 7 while bits 0 thru 2 provide the "Mode Field" seen later in Fig. 4.

In Figure 3, byte "2" is the buffer identification designated "Buffer ID". Bytes "3" thru
15   "5" are the buffer offset area for data which encompasses the most significant bit (MSB) to the least significant bit (LSB).

In Figure 3, the bytes "6" thru "8" provide for the parameter list length and also indicate the most
20   significant bit to the least significant bit. Then byte "9" is the control statement which is the final informational portion of the Write Buffer Command.

The function of this Write Buffer Command and the meaning of the fields within the Command Descriptor
25   Block (Fig. 2) depend on the contents of the "Mode" field, Fig. 4.

Referring to Figure 4, there is seen a diagram of the Write Buffer Mode field. It should be noted that when the system is downloading microcode with buffer
30   offsets, then the Write Buffer Command Mode should be 110b or 111b (Fig. 4).

Referring to Figure 4, there is seen the mode 000b designated as the Write Combined Header and Data Mode. In this mode, data to be transferred is proceeded
35   by a four-byte header. The four-byte header consists of all reserved bytes. The buffer ID in the buffer offset

awk\appl\041468L.doc

fields are made to "zero." The parameter list length field specifies the maximum number of bytes that are to be transferred from the Data Out Buffer. This number includes four bytes of header so that the data length to

5 be stored in the device buffer is the parameter list length minus four. If the parameter list length exceeds the buffer capacity, then the device service must return "Check Condition" status and sets the sense key to "Illegal Request."

10 The next mode seen in Figure 4 is the 001b or vendor-specific mode. Here, the meaning of the buffer ID, the buffer offset, and the parameter list length fields are not specified, but are determined by Vendor-data.

15 The data mode 010b entitled "Write Data" indicates that the Data-Out Buffer contains buffer data destined for the logical unit of a selected target. The buffer ID field identifies a specific buffer within the logical unit. The vendor assigns buffer ID code to

20 certain buffers within the logical unit. The buffer ID "zero" is supported here. If more than one buffer is supported, then additional buffer ID codes must be assigned contiguously beginning with "1" (one). If an unsupported buffer ID code is selected, the device must

25 then return "Check Condition" status and then sets the sense key to "Illegal Request" with additional sense code of "Invalid Field In CDB."

Data is written to the logical unit buffer starting at the location specified by the buffer offset.

30 The application client must conform to the offset boundary requirements returned in the Read Buffer descriptor. If the device server is unable to accept the specified buffer offset, it must return Check Condition status and then sets the sense key to Illegal Request

35 with an additional sense code of Invalid Field In CDB.

awk\appl\041468L.doc

The parameter list length specifies the maximum number of bytes to be transferred from the Data Out Buffer to be stored in the specified buffer beginning at the buffer offset. The application client must attempt
5 to ensure that the parameter list length plus the buffer offset does not exceed the capacity of the specified buffer. The capacity of the buffer may be determined by the buffer capacity field in the Read Buffer descriptor. Here again, if the buffer offset and parameter list
10 length fields specify a transfer in excess of the buffer capacity, the device must return a status of Check Condition and then sets the sense key to Illegal Request plus the additional sense code of Invalid Field In CDB.

It should be noted that the present embodiment
15 with double two-dimensional arrays provides an unusually large capacity buffer allowing firmware (= 393,216 bytes or less) download in a single command cycle instead of multiple download cycles.

The next level in the Write Buffer mode field
20 is the Download Microcode Mode, 100b. Now if the logical unit (target) cannot accept this command due to some device condition, then the device server must terminate each Write Buffer command with this mode 100b and a Check Condition status, plus a sense key of Illegal Request.
25 It then sets the additional sense code to read Command Sequence Error. In this mode, the vendor-specific microcode or control information is transferred to the control memory space of the logical unit. After a power-cycle or reset, the device operation will revert to a
30 vendor-specific condition. The meanings of buffer ID, buffer offset, and parameter list length fields are not presently specified and are not required to be filled with zeros. When the microcode download has completed successfully, the device will generate a unit attention
35 condition for all initiators except the one that has issued the Write Buffer command. Then an additional

sense code will be set to indicate Microcode Has Been Changed.

      Now referring to the Write Buffer mode field at Figure 4, the next item is the mode 101b designated

5  Download Microcode and Save. If the logical unit (target) cannot accept this command because of some condition, the device server will terminate each Write Buffer command with this mode designated 101b and with a Check Condition status, plus a sense key of Illegal

10  Request and then will set the additional sense code to Command Sequence Error. In this mode, any vendor-specific microcode or control information is to be transferred to the logical unit, and if the Write Buffer command is completed successfully, will also be saved in

15  a non-volatile memory space, such as a semi-conductor, a disk or other medium. The downloaded code will be effective after each power cycle and reset until it is supplanted in another download microcode and save operation. The meanings of the buffer ID, buffer offset,

20  and parameter list length fields are not required to be zero-filled. When the Download Microcode And Save Command has completed successfully, then the device will generate a unit attention condition for all initiators except the initiator that issued the Write Buffer

25  command. When reporting the unit attention condition, the device will set the additional sense code to Microcode Has Been Changed.

      The next item in Figure 4 is the mode 110b designated as Download Microcode With Offsets. In this

30  mode the application client may split the transfer of the vendor-specific microcode or control information over two or more Write Buffer commands. If a logical unit cannot accept this command because of some device condition, the device will terminate each Write Buffer command with

35  this mode, 110b, with a Check Condition status, also a

sense key of Illegal Request and set the additional sense code to Command Sequence Error.

If the last Write Buffer command (of a set of one or more commands) completes successfully, the microcode or control information is transferred to the control memory space of the logical unit. After a power-cycle or a reset, the target device shall revert to a vendor-specific condition. In this mode, the Data-Out Buffer contains vendor-specific, self-describing microcode control information.

Since the downloaded microcode or control information is sent using several commands, then when the logical unit detects the last Download Microcode With Offsets And Save mode Write Buffer command which has been received, the device will perform any logical unit required verification of the complete set of downloaded microcode or control information prior to returning Good Status for the last command. After the last command completes successfully, the device server will generate a Unit Attention condition for all initiators except the one that issued the set of Write Buffer commands. When reporting the Unit Attention condition, the device will set the additional sense code to "Microcode Has Been Changed."

If the complete set of Write Buffer commands required to effect the microcode or control information change (one or more commands) are not received before a reset or power-on cycle occurs, then the change shall not be effective and the new microcode or control information will be discarded.

The buffer ID field (Byte 2 of Write Buffer Command, Fig. 3) identifies a specific buffer within the logical unit. The vendor assigns buffer ID codes to the buffers within the logical unit. A buffer ID value of zero will be supported. If more than one buffer is supported, additional buffer ID codes are assigned

awk\appl\041468L.doc

contiguously beginning with one. If an unsupported buffer ID code is identified, then the device will return the status of Check Condition and will also set the sense key to Illegal Request with an additional sense code of

5 Invalid Field In CDB.

The microcode or control information is written to the logical unit buffer starting at the location specified by the buffer offset. The application client will send commands that conform to the offset boundary

10 requirements. If the device is unable to accept the specified buffer offset, it will then return the Check Condition status and set the sense key to Illegal Request with the additional sense code of Invalid Field In CDB.

The parameter list length specifies the maximum

15 number of bytes that must be present in the Data-Out Buffer to be stored in the specified buffer beginning at the buffer offset. The application client must attempt to ensure that the parameter list length plus the buffer offset does not exceed the capacity of the specified

20 buffer. If the buffer offset and parameter list length fields specify a transfer in excess of the buffer capacity, the device will return the status of Check Condition and set the sense key to Illegal Request plus the additional sense code of Invalid Field In CDB.

25 With the expanded double two-dimensional array of Fig. 6, there is no longer any hang-up over buffer capacity which is now usually more than enough capacity. The final line in the Write Buffer mode field of Figure 4 is designated as 111b and indicates the operation

30 designated Download Microcode With Offsets And Save mode. In this mode, the initiator may split the transfer of the vendor-specific microcode or control information over two or more Write Buffer commands. If the logical unit cannot accept this command because of some device

35 condition, the device will terminate each mode 111b Write Buffer command with a status of Check Condition plus the

sense key of Illegal Request plus the additional sense code to designate Command Sequence Error.

If the last Write Buffer command of the set of one or more commands completes successfully, the

5   microcode control information is saved in a non-volatile memory space such as semi-conductor or disk or other target media.  Now since the downloaded microcode or control information may have been sent using several commands, then when the logical unit detects the last

10  download microcode with offsets and save mode Write-Buffer commands that have been received, the device server performs any logical unit-required verification of the complete set of downloaded microcode or control information prior to returning the status of "Good" for

15  the last command.

The buffer ID field identifies a specific buffer within the logical unit, as discussed hereinbefore.

The parameter list length also specifies the

20  maximum number of bytes that can be present in the Data-Out Buffer to be stored in the specified buffer beginning at the buffer offset.  As before, if the buffer offset and parameter list length field specify a transfer in excess of buffer capacity, the device will return the

25  Check Condition status, as described hereinbefore.

Figure 5 is a diagram which lays out the Operation Code.  As seen, the Operation Code uses bits 0-4 for the Command Code and bits 5 thru 7 for the Group Code.  All devices implement their commands with a

30  mandatory operation code in the appropriate section for their device types.  For example, a command is executed by sending a Command Descriptor Block (CDB) to the target logical device.  For each of the Command Descriptor Blocks (CDB, Fig. 2), it will be noticed that the first

35  byte (bytes) of the CDB will be the Operation Code while

the last byte (byte 11) of the CDB will be the Control Byte.

Thus, each Operation Code has a Group Code field and a Command Code field. It will be noted that

5   the three-bit Group Code field provides for 8 groups of command codes while the five-bit Command Code provides for 32 Command Codes for each Group.

Thus, as seen in Figure 5 under the Group Code field, group zero has 6 command bytes, group 1 has 10

10  command bytes, group 2 has 10 command bytes for a new version of SCSI-2, while groups 3 and 4 are reserved, and group 5 has 12 command bytes, while 6 and 7 are specific to a given vendor as to the number of command bytes to be involved.

15  Fig. 6 is a drawing which illustrates the two-dimensional array involved in the SCSI download. This is actually implemented doubly to establish two of the two-dimensional arrays. Each array will be seen to have a number of rows; for example, this may be 10 rows. Then

20  each row will have a number of elements for digital units, for example, this may be 20. This type of dimensional array would be described as A[0:9,0:19] which would indicate that there are 10 rows designated 0 through 9, and each row has 20 digital units designated

25  as 0:19. At least two of such two-dimensional arrays are provided in this embodiment which allow the download cycle in one operation rather than a series of concatenating cycles.

Fig. 6 is an illustration of a two-dimensional

30  array, which in this example is seen to hold 393,216 bytes.

Referring to Fig. 6, the array buffer which is designated as [0:47, 0:8192] is shown to indicate that there are 48 row numbers numbered from 0 thru 47 and at

35  the same time there are 8,192 column numbers.

The array shows a series of blocks or slots, each of which can hold an eight-bit byte of information. As a result, since there are 48 rows and each row has 8,192 bytes, then the two-dimensional matrix array will provide a totality of 393,216 bytes of information.

Fig. 7A is a flow chart indicating the steps involved for implementing the specialized download operation to a particular SCSI target. Referring to Fig. 7A, the first step (i) is designated DFAST which is a particular utility program used to download firmware for SCSI targets. This program thus downloads microcode/firmware to SCSI devices, such as disk drives or tape drives which are attached to a Unisys A-Series computer system.

The second step (ii) involves entering the name of the firmware file involved so that this file can be accessed from memory.

At step (iii) the sequence will then fetch this particular firmware file, after which it can proceed to step (iv).

Step (iv) is a decision block as to whether this firmware file will be downloaded to the peripheral controller or not. If this is to be downloaded to the controller (YES), then the sequence proceeds to step (iva). If the firmware file is not to be downloaded to the controller (NO), then the sequence proceeds to step (v) which is another decision block which poses the question whether the firmware is to be downloaded to the target device. If the answer here is (YES), then the sequence proceeds to reference mark A. If the answer is (NO) at step (v), then the sequence proceeds to step (vi) which is a decision block as to whether there should be a verification of the firmware only. If the answer here is (YES), then the procedure goes to mark B. If the answer is (NO), then at step (vii), the program will terminate at this point.

awk\appl\041468L.doc

Referring to Figure 7A at step (iv) which involves a decision point whether to download to the controller. Here, if the answer is (YES), then the sequence proceeds to step (iva) where there is a need to
5   display and enter the "controller ID" to select the appropriate peripheral controller. The next step is step (ivb) which involves the assignment of the controller, after which there is a decision block. At step (ivc) there is involved a question as to whether the peripheral
10  controller has been assigned for utilization.

If the answer is (YES), that is to say a controller has been assigned, then the sequence proceeds to step (ivd) which involves a reading of the controller attributes.
15  The next step is (ive) where there is a comparison made to compare the firmware header file with the expected target attributes. After this, at comparison step (ivf), a decision block is reached as to whether the firmware header file is the same as the
20  target attributes. Here, if the answer is (YES), then the sequence precedes the step (ivg) which involves another decision block querying whether the sequence will still want to download. If the answer here is (YES), then the procedure continues on via reference mark E to
25  Figure 7D.

Returning back to step (ivc), if there has not been an assignment of a controller (NO), then the sequence proceeds to step (ivcn) in order to recycle back to step (iva) to make sure that a controller is reserved
30  for this function.

Now, returning to step (ivf), if the compared file firmware is not the same as the target firmware (NO), then the next sequence is indicated in step (viii), i.e., to Terminate.
35  Now looking at the decision block at step (ivg), if at that point there is a need to download,

(YES) then the procedure continues on via reference mark E, which is shown in Figure 7D.

Now referring to Figure 7D at step (ivg1), the system will set up the buffer arrays for enabling a download. Then at step (ivg2), the sequence will start to download of the firmware to the buffer array, such as was indicated in Figure 6, which can utilize double sets of two-dimensional arrays.

In this particular case of Figure 7D, the step (ivg3) will use a one dimensional buffer array. Then at step at (ivg4), the controller will issue the write buffer command showing the total bytes of data involved which then will continue at reference marker M over to Figure 7C. The one-dimensional buffer array is selected for use when the firmware is less than 393,216 bytes of data which is the system max byte capacity. The two-dimensional buffer array is selected when the firmware is greater than the system max byte capacity of 393,216 bytes.

Thus from Figure 7D to Figure 7C, the reference marker M will continue on to step (vq4), where a Test Unit Ready Command is issued and then a step (vq5) query to check if the Test Unit is ready? If YES then step (vq6) will read the Inquiry Data (attributes) and display them at step (vq7) after which at step (vq8), an inquiry command is used. Then step (vq9) displays the servo inquiry data and continues via mark C to Fig. 7A. Here, steps (iv) through (viii) operate as previously described.

awk\appl\041468L.doc

Now returning to Figure 7A at the step (vi) which involves a decision block as to whether to verify only the firmware. If the answer here is YES, then the sequence proceeds via reference marker B over to Figure

5  7D. Here the next step (via) is where there will be a display of entering the name of the firmware file. After this at step (vib), there will be a retrieval of the firmware file, after which at step (vic) there will be a display of the firmware file attributes for the user to

10  provide visual validation. This then proceeds via reference marker C over to Figure 7A at step (iv).

Now referring to Figure 7B, it should be noticed that step (v) having the "YES" leg of Figure 7A is now continued on Figure 7B at the reference marker A.

15  Here then there occurs step (va) where there is entered and displayed the device ID (identification). Then at step (vb), there is an assignment of the device which is the target device.

The next step (vc) is a decision block as to

20  whether or not the target device has been assigned. If the answer is (NO), then at step (vcn) the sequence returns the step (va) in order that there be a certainty that the device is reserved through a display (vcn).

At step (vc), if the device has been assigned

25  (YES), then the sequence proceeds to step (vd) in order to read the device Inquiry Data. Then at step (ve), a comparison is made between the firmware header file as against the target attributes which results in a decision block at step (vf) where the comparison, when it provides

30  a "YES" result, will then go to step (vg) for another decision block to decide whether the new firmware is the same as the old firmware and whether there is a necessity to still download the new firmware. Here, if the answer is (YES), then at step (vh) the system will set up the

35  buffer arrays for the downloading.

The next step is step (vI) which will read the inquiry servo data, after which at step (vj) there will be a comparison of the servo firmware header data with the expected target inquiry servo data. The servo
5    firmware header revision level is compared with the expected target Inquiry Servo data to see if the system has already upgraded the firmware. If they compare, then the User is asked whether he still wants to reload the same level of servo firmware.

10    Then at step (vk), there is a comparison made, which if the answer is "YES" will then to proceed to step (vL) to enable a decision block to determine whether the new servo firmware is the same as the old and whether there is still a need to download. If there is a YES
15    decision to further download, then the sequence proceeds to step (vm) which starts the download of firmware which then sequences to a new decision block at step (vn). This step queries whether the firmware is greater than 393,216 bytes. Here if the answer is "YES", then step
20    (vo) the system will use the first two-dimensional buffer array. In Fig. 7B, step vo, and in Fig. 7C, step vq1, a test is made to see if the firmware is greater than 393,216 bytes, and if it is, the selection will be made to use the two dimensional buffer array.

25    Then at step (vp), the sequence will issue a write buffer command with 8,192 bytes of data. It is possible to control or select the size or length of each block being transferred within the limits of the hardware and software specifications. Even though a User could
30    put up to 4,294,967,295 blocks/bytes of data, most systems have smaller buffer sizes that could not use all of the four bytes allocated for a typical 12 byte CDB. The Write Buffer command is a 10 byte CDB and only allocates 3 bytes for the parameter list length which is
35    a maximum of 16,777,215 bytes/blocks. In the Unisys A-Series system, a total of 393,216 bytes of data would

be the maximum that could be transferred in a Write Buffer command. When firmware data is greater than the system capacity of 393,216 bytes, the data is transferred in 8,192 byte blocks per Write Buffer command which

5 complies with the Firmware Download Specification. Then at step (vq), a decision block is reached which will query whether there has been 393,216 bytes having been downloaded, which is the limit of the first two-dimensional buffer array.

10 Then at the reference marker I which pertains to Figure 7C, (from Figure 7B to Figure 7C), the sequence then proceeds to step (vq1) where the sequence will use the second two-dimensional buffer array. Then the sequence proceeds to (vq2) where a write buffer command

15 will be issued with 8,192 bytes of data. Then at step (vq3), a decision block queries whether the download is complete. If the answer is YES, then the sequence proceeds to step (vq4) which will then issue a TEST UNIT READY command. Then at step (vq5), a decision block will

20 query whether the test unit is ready. If the answer is "YES," then the sequence proceeds to step (vq6) in order to read the attributes of the file. Then at step (vq7), there will be a display of the attribute data, after which at step (vq8) there will be an Inquiry Command

25 issued which then proceeds to step (vq9) where there is a display of the Servo Inquiry Data. This then proceeds via reference marker C of Fig. 7A. This sequence then proceeds into step (iv) which was described previously.

Subqa It may be noted that steps (vf), (vg), (vh) and

30 (vl) have a NO sequence which continues via mark J to Fig. 7A at step (viii), Terminate.

On step (vn) of Fig. 7B, the NO leg shows marker H to Fig. 7C where step (vn1) will select use of the one-dimensional array and then at step (vn2) will

35 issue a Write Buffer Command specifying the total bytes of data to use. Then marker M continues to step (vq4),

awk\appl\041468L.doc

Fig. 7C, which continues on the previously described sequence (vq4) through (vq9) then (iv) through (viii).

Described herein is a system and method for more efficiently downloading firmware microcode to a
5    target controller which manages a disk or tape unit. The system queries the target to find its attributes so that the proper firmware will be downloaded in an efficient manner. This involves selecting a single buffer array, two-dimensional buffer array or a double two-dimensional
10   buffer array. Additionally, a check is made of the prior firmware to verify whether or not the newly to-be-loaded firmware is different from the prior firmware so as to eliminate any need for download if the prior and new firmware is found to be exact duplicates of each other.

15   While a preferred embodiment of the disclosed invention has been described, there may still be other implementations which involve the concept of the present invention which is defined in the attached claims.

awk\appl\041468L.doc